Peer-reviewed & Open access journal
www.academicpublishingplatforms.com

ATI - Applied Technologies & Innovations
Volume 4 | Issue 1 | April 2011 | pp. 48-57

Methods and algorithms of adaptive designing for neuronetworking system of processing the data with non-stationary nature

Olimjan Djumanov
Faculty of Mathematics, Samarkand State University, Uzbekistan
e-mail: s-xolmonov@mail.ru

The paper examines hybrid model of data neuronetworking processing system, in which the opportunities of a wide spectrum of methods and algorithms of neural network and statistical models are combined. The original ways and algorithms of adaptation during designing of neural network are developed by escalating, selection and adjustment of activation functions. The strategy and new principles are developed for neural network output quality control. The theoretical results were approved on the basis of Koxonen model; and as result conditions of well over reduction of output data approximation error are proved.

Keywords: Architecture of neural network, adaptation, nonstationarity, activation function, recognition of micro objects.

Introduction

The development of methods and algorithms combining opportunity of statistical methods and neural networks (NN) models for construction of adaptive processing systems of information with non-stationary nature represents an imperative direction of scientific research. The necessity of such approach is especially shown at the decision of applied tasks of microobjects images visualization, recognition and classification, analysis and forecasting and interpretation of the data, connected to development of new specific ways and algorithms of: adaptations of NN training at the expense of account the data statistical and dynamic properties; organization the ways of division the multidimensional input vector on some vectors of smaller dimension for formation an optimum NN training sequence; parallel processing of the information for reception of final result with smaller computing expenses (Widrow and Lehr, 1990).

The present work develops within the limits of the hybrid model and offers the information neuronetworking processing system in which a wide spectrum of possibilities of NN methods and algorithms and statistical models are combined. Working out of adaptive algorithms of training is directed on formation of optimization ways for training subset with minimum error. Problems of NN training are solved by taking into account error in the data, finding out the optimum size of training sequence on the basis of use of ways of classification and knowledge extraction from the data with dynamic nature.

Traditionally, neuronetworking data processing system assumes three main tasks: selection of informative signs; object modeling; decision-making and producing of operating acts. The system on the basis of hybrid model carries out functions of preliminary processing of the information on the basis of analysis statistical methods, creation of rational training sequence, designing of NN structure, optimization of objects model, updating of parameters and escalating of NN, adaptation of training parameters on the basis of algorithms of processing the data with non-stationary nature.

The author considered in previous research the study methods and adaptive algorithms of training of neuronetworking system of processing the data with non-stationary nature (e.g., Djumanov, 2007, 2008, 2009). These works examined particularly: the information statistical and dynamic characteristics; choice of the most informative signs from heuristic initial set (ensemble); use the ways of classification and knowledge extraction from the data with dynamic nature. These works stated the developed techniques of an informatively signs estimation by means of the device of regression-correlation analysis through calculation of factors of pair and plural correlation; the determined and multi-factor analysis with definition of coefficient of influence and elasticity of factorial indicators on productive, coefficients of plural correlation used as a measure of data informatively estimation.

The data of statistical parameters obtained by this way were used in the construction of subsystem preprocessing of information for selecting adequate model of teaching NN with different architecture and for designing adaptive procedures and improving of convergence in teaching NN.

Below we present the results of researching in development the adaptation the ways and algorithms of the learning in the construction neural network processing of information.

Adaptation in designing neural network

It is demanded to define quantitative valuation of sign *i* as the sum of the whole *i*- th sum as neural input of the first layer NN. But for the assessment of the valuation NN's input we must appreciate not only the valuation of neuron but the valuation neuron output and other layers of neuron until last layer.

And with it, for NN which has one output, that is with one neuron in the last layer, the influence of *i*-th sign x_j - the value of which come to the input of NN first layer - to output signal *y* will be defined by the input signs weight value. In connection with it we worked out the adaptation of algorithims in desingning NN which based on the rule to set the weights moduluses and NN threshold with a glance of mistake in transformation of each neuron. Offering algorithm includes the next steps:

Step 0. Keeping the current collection of neuron weights and thresholds values.

Step 1. Calculating the value of weights $\overline{w}$ meaningfulness by the given rule. In the capacity of this value we used the average arithmetical value of the net weight and thresholds modulus.

Step 2. All weights meet condition $\left|w_j^{(\mu,i)}\right| < \overline{w}$ decrease by rule $w_j^{(\mu,i)} = R^-(w_j^{(\mu,i)})$,

where $w_j^{(\mu,i)}$ - is the weight of j-th input of i-th neuron of μ- th NN layer; but μ, i, j are determined by the rules set beforehand .

Step 3. All the weights, meet conditions $\left|w_j^{(\mu,i)}\right| \geq \overline{w}$, increase by rule $w_j^{(\mu,i)} = R^+ w_j^{(\mu,i)}$.

Step 4. Calculating for all copies of training sample x^s, $s = 1,2,\dots,s$ the Error - value of the net total error.

Step5. If Error<MaxError, where MaxError - is the most bounded valuation of the net error, then go to step 0, else go to step 6.

Step 6. Restoring the previous valuation of the net weights and thresholds.

Step 7. End.

The rules which were offered for building of adaptive algorithm in NN constructing and identified as R(w), will set as below.

а) $R^{-}(w) = \begin{cases} w-\alpha, & if \quad w > 0, \quad w \geq \alpha; \\ w+\alpha, & if \quad w < 0, \quad w \leq -\alpha; \\ 0, \ if \ (w=0) \ or \ (w>0 \, and \ w<\alpha) \ or \, (w<0 \text{ è } w>-\alpha). \end{cases}$

б) $R^{+}(w) = \begin{cases} w+\alpha, & if \ \ w > 0; \\ w-\alpha, & if \ \ w < 0; \\ 0, & if \ \ w = 0. \end{cases}$

or

а) $R^{-}(w) = \begin{cases} w\alpha, \ 0 < \alpha < 1, \ if \ \ w > \xi; \\ 0, \ \ if \ \ w \leq \xi. \end{cases}$

б) $R^{+}(w) = w / \alpha, \ 0 < \alpha < 1,$ where α , ξ - the constants which were given beforehand.

or

а) $R^{-}(w) = w(1 - P(w));$

б) $R^{+}(w) = w / P(w)$,

where $P(w) = \begin{cases} -\dfrac{2}{\pi} arctg \dfrac{1}{w}, \ if \ \mathrm{w} < 0; \\ \dfrac{2}{\pi} arctg \dfrac{1}{w}, \ if \ \mathrm{w} > 0; \\ 1, \ if \ w = 0, \end{cases}$

We must mention that because of the offered algorithm the first phase of NN's architecture constructing are traced as moving off useless connections; but we will have to investigate in future the rationality of this construction.

In the next section of our issue, we offer some complex of original procedures for adaptation of algorithm in NN constructing which based on increasing the net.

Increasing the neural network

This algorithms includes the next steps;

Step 0. Initialization: $\mu = 1$.

Step 1. If the number N_{μ} of neurons in μ-th NN layer less than certain size β, then increase the number neurons in μ-th NN layer by the rule $N_{\mu} = N^{+}(N_{\mu})$, where $N^{+}(N_{\mu})$ may be identified by one of the below:

$$N^{+}(N_{\mu}) = \begin{cases} N_{\mu} + \gamma, & if \quad (N_{\mu} + \gamma) < \beta; \\ \beta, & if \quad (N_{\mu} + \gamma) \geq \beta, \end{cases}$$

or

$$N^{+}(N_{\mu}) = \begin{cases} \text{round } (N_{\mu}\gamma), \gamma > 1, & if \quad \text{round } (N_{\mu}\gamma) < \beta; \\ \beta, \; if \quad (N_{\mu} + \gamma) \geq \beta, \end{cases}$$

Where β and γ - constants which were set beforehand.

Go to step 4, else go to step 2.

Step 2 . If $\mu < M$, then accept $\mu = \mu + 1$ go to step 4, else go to step 3.

Step 3. If $\mu = M$ and $M < M_{\max}$, where $M_{\max}$ - is the most admissible number of NN layers, then to accept: $M = M + 1$, $N_M = 1$ and go to step 2 , else go to step 4.

Step 4. End.

We must say that this way of NN increasing determines the issue to select rational coefficient of adaption in the process oj learning. In the results of researching it was formed the next condition of NN installation by setting constants β and γ:

1. in increasing the valuation of γ, the net will grow faster;
2. the valuation of β and γ must be more than zero, constant β must be integer;
3. if constant γ - is not integer, then the value of $N^{+}(N_{\mu})$ must be rounded;
4. with the increasing the value of β will decrease the facilities in increasing of the net.

The originality of this way of adaption in NN constructing is the decision in choosing the structure of NN without effortful procedure for rating of architecture rationality.

Showing the obtaining results we affirm that alongside with getting the effective NN training set and desining the rational architecture of NN - the bullet point in constructing of neuronetworking data processsing is the decision in choosing the adeqaute activation function and in working out the strategy of NN output quality control.

In connection with it, the next section of paper is devoted to the working out the adaptive NN learning algorithm which based on installation of activation functions.

Learning of NN by the way of adaptation the activation function

The principle of algorithms construction concludes the adaptation of the activation function centre and width of their windows, working out the procedures of checking up the adequacy the activation function within the limits of chosen window on training set (practice sample), rating the mistake of approximation which doesn't coincide with the centre of activation function.

Algorithm includes the following steps.

Step 1. Designating the size of hidden layer H is equal to the number of practice samples Q, synaptic weights of hidden layer neurons is equal to 1.

Step 2. Fixing the centre of activation function of hidden layer neurons in the points within input signals of the net which are the part of the practice samples : $c_j = x^j$, $j = 1,2,...,H$, where x^j - is the vector of signs of j-th copy in chosen training sample.

Step 3. Choosing the width of windows for hidden layer neurons activation function σ_j, $j = 1,2,...,H$ rather big, but they must not lay on each other in the space of input signals of the net.

Step 4. Defining the weight of exit layer neuron w_{jj}, $i = 1,2,...,Z$; $j = 1,2,...,H$. For this we produce the whole collection of the practice samples.

The output of i-th neuron from output layer for p-th sample is equal:

$$D_j = w_{i1} f(x^p, c_1) + w_{i2} f(x^p, c_2) + ... + w_{iH} f(x^p, c_H) =$$
$$= w_{i1} f(x^p, x^1) + w_{i2} f(x^p, x^2) + ... + w_{iH} f(x^p, x^H).$$

where $f(\cdot)$ - is the activation function of approximation conformable to input and output vectors.

Through demonstrating this equation for the all net output and for the all samples we obtain the next equation in the form of matrix:

$$\Phi w^{\mathrm{T}} = D$$

where $\Phi = \begin{pmatrix} f_{11} & \cdots & f_{1\mathrm{H}} \\ ... & ... & ... \\ f_{\mathrm{H}1} & \cdots & f_{\mathrm{HH}} \end{pmatrix}$ - interpolating matrix, $f_{ij} = f(x^i, x^j)$;

$w = \begin{pmatrix} w_{11} & \cdots & w_{1\mathrm{Z}} \\ ... & ... & ... \\ w_{\mathrm{H}1} & \cdots & w_{\mathrm{HZ}} \end{pmatrix}$ - matrix of output synaptic weights;

$D = \begin{pmatrix} D_{11} & \cdots & D_{1\mathrm{Z}} \\ ... & ... & ... \\ D_{\mathrm{H}1} & \cdots & D_{\mathrm{HZ}} \end{pmatrix}$ - the matrix of the output samples.

Decision of $w^{\mathrm{T}} = \Phi^{-1} D$ gives us required values of output synaptic weights, providing the passing interpolated surface through the practice samples.

The theoretical results written above were tested based on the NN multilayer models, radial basic net and Kohonen's model. Below we'll show the results on realization the theoretical suggestions based on the Kohonen's model (Kohonen, 1990).

Adaptation of NN learning on model by Kohonen

The space for the tasks on NN adaptation which were written above will be widen by the working out the strategy and new principles of checking up the NN output quality which were used on Kohonen's model.

We determined that the Kohonen's model gives us less mistake in approximation of the output datas when we have the tasks microobjects visualization, recognition and classification, where as microobjects we can concider pollen-grains, fingerprints, blood cells (Banarse, 1997, Basit Hussain et al., 1994). For example, let the number of input vectors in the training set of NN be greater than chosen number of clusters; and it is necessary to determine the centre of clusters distributed in the input space. In this case, for approximation of the function of density of output vectors probability, Kohonen's algorithm will form effectively the map of the signs and the size of training subset will decrease during the process constantly.

General principle of quality control in NN output is consisting of the next. We must calculate the distance (metrics) d_j between input vectors and each output nodes j. The node j^* will be defined with minimal distance d_j, the weight of the input of nodes will be corrected in the field of the nodes j^* so that the new value of the weights must be equal.

$$w_{ji}(t+1) = w_{ji}(t) + \eta(t)(x_i^s - w_{ji}(t)), \quad j \in NE_j(t), \ i = 1,2,...,N .$$

In this case the adaptation (correction) increment $\mu(t)$ $(0 < \eta(t) < 1)$ will be defined, and it must decrease with growth of t .The character of carrying out investigation for recognition and classification of microobjects allow us to work out various variants of modified Kohonen's NN learning algorithm.

Adaptive learning algorithm A

We take $x(t)$ - as input sample from the great number of the continuous information that is the covers of microobjects, $m_i(t)$ - is the presentation of uninterrupted sequence m_i, discretizing by time. Beginning from the initial defined values, the main process of algorithm will define the next expressions:

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)],$$

if x and m_c belong to the same class they will be define by:

$$m_c(t+1) = m_c(t) - \alpha(t)[x(t) - m_c(t)],$$

if x and m_c belong to the different classes they will be define by:

$$m_i(t+1) = m_i(t), \ \forall i \neq c .$$

Adaptation coefficient $\alpha(t)$ may be constant or will decrease monotonously with lapse of time. Here is $0 < \alpha(t) < 1$.

For the algorithm A, will be recommended the initial value of α must be less than 0.1.

Adaptive learning algorithm B

The principle of NN learning by algorithm includes the next. Two vectors with independent parameters m_i and m_j, are the closest neighbours x and modify at the same time. It supposed that one of them must belong to the class 1 and the other to the class 2. In addition, x must be situated in the zone of the „windows" values, which was determined about the middle or the flatness formed by the vectors m_i and m_j.

Let d_i and d_j - Euclidean distance x from m_i and m_j. Then x get to the „window" with concerning width w, if

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s, \text{ where } s = \frac{1-w}{1+w}.$$

We recommend that the value of concerning width w of the "window" must be within from 0.2 to 0.3.
The output of adaptive learning algorithm has the appearance:

$$m_i(t+1) = m_i(t) - \alpha(t)[x(t) - m_i(t)];$$

$$m_j(t+1) = m_j(t) + \alpha(t)[x(t) - m_j(t)],$$

Where m_i and m_j - are two closest to x vectors with independent parameters, and x with m_j belong to the same class, but x and m_i belong to the different classes, x must get to the „window".

Adaptive learning algorithm C

The principle of NN learning by this algorithm is based on the idea putting the changes which guarantee that m_i continue approximation in the distribution of classes.
Modified rule of this algorithm were written as:

$$m_i(t+1) = m_i(t) - \alpha(t)[x(t) - m_i(t)],$$

$$m_j(t+1) = m_j(t) + \alpha(t)[x(t) - m_j(t)].$$

If x, m_i and m_j belong to the same classes, then

$$m_k(t+1) = m_k(t) + \varepsilon\alpha(t)[x(t) - m_k(t)] \text{ for } k \in \{i, j\},$$

where m_i and m_j - two the closest to x vector with independent parameters . And x with m_j belong to the same class , but x and m belong to the different classes and x must get to the „window" .
It was determined by us that the values of adaptive coefficient ε must be placed between

0.1 and 0.5, but optimal value of ε depends on the size of the least window.

Adaptive learning algorithm A1

The principle of NN learning by this algorithm is based on the modification of algorithm A thereby each m_i will be fixed individual rate of learning $\alpha_i(t)$.

Then the discretizing by time process of learning will be determined by equation

$$c = \arg\min_i \{\|x - m_i\|\},$$

where the output of algorithm will be defined by rule

$$m_c(t+1) = m_c(t) + \alpha_c(t)[x(t) - m_c(t)].$$

If x was classified correctly we will have

$$m_c(t+1) = m_c(t) + \alpha_c(t)[x(t) - m_c(t)].$$

If x was classified not correctly we'll have

$$m_c(t+1) = m_i(t), \quad \forall i \neq c.$$

And the value of adaptive coefficient $\alpha_i(t)$ is defined by recursive formula

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t)}.$$

Thereby worked out algorithms are self-stabilized and the optimal placement of m_i does not change during the long learning.

Experimental research

The rating of effectiveness offered models and methods were carring out by the datas of decision the task of the pollen-grain recognition and classification (Boucher et al., 2002). We carried out comparative analysis the effectiveness of NN with various architecture on the base of personal computer with processor Intel Pentium 4 in MatLab 7.7 under OS Linux 2.6.28. In the hidden layer as the function of activation the hyperbolical tangent was used and in the output layer the line function was used.

The learning of the net fulfilled on the base of algorithms A, B, C, A1. Efficiency of the tested NN were rating by the following criteria: the time of calculation in seconds T and the average quadratic mistake ε. The learning of the net was carried out on sample with $2 \cdot 10^4$ casual points, evenly distributed in the space [-1;1]. The process of the learning was completed with achieving the given level of the accuracy $\varepsilon = 0{,}0001$. Below we present comparative efficiency of the calculation (Table 1).

TABLE 1. COMPARATIVE EFFICIENCY OF NEURAL NETWORKS

Modified algorithm models by Kohonen	Time of delay of NN (sec.)	Error of the NN output function
Algorithm A	47.96	14.825
Algorithm B	208.84	12.846
Algorithm C	78.77	12.778
Algorithm A1	296.32	7.432

Table 2 shows the results of comparative analysis for efficiency of worked out learning algorithms in neuronetworking data processing.

TABLE 2. COMPARATIVE ANALYSIS OF LEARNING ALGORITHMS EFFICIENCY

Modified algorithms by Kohonen model	Probability of the correct classification in various experiments(in %)							
Algorithm A	100	88	85	94	94	86	82	97
Algorithm B	100	92	93	95	93	88	91	98
Algorithm C	100	93.7	93.7	97	96	92	93.7	98
Algorithm A1	100	95.6	95.1	98	97.8	96	97	100

Conclusion

Worked out algorithms of NN synthesis with rational configuration allow us to shorten the space of the training sets and by this decrease the demand to computer resources, increase the speed of work of the classifiers and allow to decrease superfluity of the model.

Worked out methods allow us to grow and shorted the structure of neuronetworking system, be able flexible adapt to the changes in condition of data process and may be recommend for using in decision the task of recognition, when we have the high demands to reliability of recognition.

During the testing adaptive NN learning algorithms based on Kohonen's model it was proved that we can achieve much less mistakes of aproximation in output datas. When the number of input vectors in the NN training set are greater than chosen number of clusters, the size of training subset decreas in time consonantly and algorithms take into account statistical and dynamical characteristich of datas.

References

Banarse, D., 1997. “A generic neural network architecture for deformation invariant object recognition,” Unpublished PhD Thesis, University of Wales, Bangor, pp.56-64.

Basit Hussain and Kabuka, M., 1994. “A novel feature recognition neural network and its application to character recognition,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 16(1), p.98-106.

Boucher, A., Hidalgo, P., Thonnat, M., Belmonte, J., Galan, C., Bonton, P., and Tomczak, R., 2002. “Development of a semi-automatic system for pollen recognition,” Aerobiologia, 18(3-4), pp.195-20.

Djumanov, O., 2007. “Programmed system for the adaptive monitoring of the continuous nature information on the basis of supervised learning of neural network,” IT Promotion in Asia, TUIT, Tashkent, Information Technology Internationalization Research Center, pp.181-90

Djumanov, O., 2008. “Adaptive training of intellectual system of processing and analysis of the non-stationary on nature information,” Fifth World Conference on Intelligent Systems for Industrial Automation, 25-27 November, Tashkent, Uzbekistan, pp.133-40.

Djumanov, O., 2009. “The adaptive control of accuracy at transfer and processing of the continuous information in view of non-stationary process,” IT Promotion in Asia, Tashkent University of IT, Uzbekistan, ITIRC, Korea, pp.116-20.

Kohonen, T., 1990. “Improved versions of learning vector quantization,” Proceedings of the International Joint Conference on Neural Networks, San Diego: IEEE, Vol.I, pp.545-50.

Widrow, B. and Lehr, A., 1990. “30 years of adaptive neural networks: Perception, madaline and Backpropagation,” In: Proc. IEEE 75 (1990), 9, 1415-442.